**Writing Comments for the JavaDoc Tool**

* unique way of commenting -- utilities read comments and generate an HTML document that has links

**JavaDoc Comments**

* Start with /\*\* and end with \*/
* Can use HTML tags within Javadoc comments
* 2 parts to a comment -- **description** and **block tags**such as @param
  + Description should start with a single sentence that describes the class, method, constructor, etc. NOTE: it ends at the first period -- if you use Mr. in the sentence, the statement ends after Mr.
* **Tips for writing Descriptions**:
  + surround keywords, class & package names, field names., etc.  
    . by <code> and </code>
  + use phrases to ensure brevity
  + used descriptive (3rd person):   
    example: retrieves the GPA instead of retrieve the GPA
  + descriptions of a method start with a VERB phrase   
    example: "updates the total credits" rather than "this method updates the total credits"
  + descriptions of fields can omit the subject   
    i.e. a label for the company name instead of "this field is a company name
* Suggestions for order of block tags
  + @param
  + @return
  + @exception
  + @author
  + @version
  + @see
  + @since
  + @serial
  + @deprecated

**2 types of JavaDoc Comments:**

1. **Class-level comments:**  
   describe the class.   
   placed above the declaration of the class.
2. **Member-level comments:**  
   describe the fields, constructors, and methods  
   may contain @return and @param lines .  
     
   **examples**  
   /\*\* returns the number of employees  
   \* @return int  
   \*/  
   public int countEmployees() {  
   ...  
   }  
     
   /\*\* updates the total number of credits  
   \* @param int additional credits taken  
   \*/  
   public void updateTotalCredits(int newCredits)  
   { this.totalCredits += newCredits;  
   }

**Tags**

|  |  |
| --- | --- |
| **@link** | Link to a class @link <className> [how to display] |
| Link to a method @link <methodHeading> [ how to display] |
| Link to a method in a different class  @link <className>#<methodHeading> [how to display] |
| **@author** | Use in class-level comments to indicate the author of the code |
| **@param** | explain a parameter of a method or a parameter in a constructor. |
| **@return.** | explain the return type of a method. |
| **@throws** | explain an exception thrown by a method. |
| **@exception** | explain an exception |

**Generating JavaDoc from Eclipse**

* GENERATING: Right-click on a method declaration, choose SOURCE
  + then Generate Element Comment
* COMPILING JavaDoc
  + Pull down on  PROJECT, and then  choose "Generate JavaDoc".
  + Look for the new folder in your project called “doc” and look for the file that is the nameOfYourClass.html to see the documentation that was created.

**Useful References/Reading**

* Oracle: [How to Write Doc Comments for the Javadoc Tool](http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html) (Required reading)
* [Requirements for Writing Java API Specifications](http://www.oracle.com/technetwork/java/javase/documentation/index-142372.html)